CONTROLLER FOR AUTOMOBILE

Patent Number: JP7277105
Publication date: 1995-10-24
Inventor(s): KURATA KENICHIRO; others: 04
Applicant(s): HITACHI LTD
Requested Patent: JP7277105
Application Number: JP19940076845 19940415
Priority Number(s):
IPC Classification: B60R16/02; G05B15/02; G06F9/06
EC Classification:
Equivalents:

Abstract

PURPOSE:To reduce development man-hours for control software and to provide a general purpose applicability in relation to a hardware change.
CONSTITUTION:Automobile control software to be described is constructed of separate components consisting of an application part 1, an I/O description part 2, and an I/O data part 3, and these are connected to each other by means of a software connecting means 4. In this way, description is carried out while the application part 1 and the I/O description part 2 are separated from each other, so that description software is simplified. As the I/O data part 3 is arranged independently of the I/O description part 2, alteration of hardware can be accomplished only by alteration of the I/O data part 3. In addition, an object program is generated by connecting these parts, and therefore the optimization process can be programmed.

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報(A)　(11)特許出願公開番号

特開平7－277105

(43)公開日　平成7年(1995)10月24日

(51)Int.Cl.[6]	識別記号	庁内整理番号	Ｆｉ	技術表示箇所
Ｂ６０Ｒ 16/02	Ｍ			
Ｇ０５Ｂ 15/02				
Ｇ０６Ｆ 9/06	５３０ Ａ	7230－5Ｂ		
		7531－3Ｈ	Ｇ０５Ｂ 15/02	Ｐ

審査請求　未請求　請求項の数5　ＯＬ　（全 11 頁）

(21)出願番号　特願平6－76845

(22)出願日　平成６年(1994)４月15日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　倉田　隊一郎
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内

(72)発明者　吉田　義幸
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内

(72)発明者　箕輪　利通
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内

(74)代理人　弁理士　小川　勝男

最終頁に続く

(54)【発明の名称】　自動車用制御装置

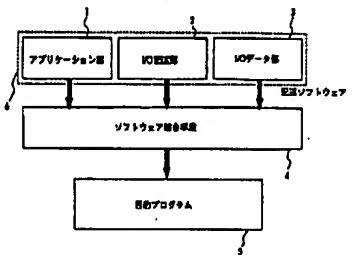
(57)【要約】

【目的】制御用ソフトウェアの開発工数を低減し、ハードウェアの変更に対して汎用性をもたせる。

【構成】記述する自動車制御ソフトウェアをアプリケーション部１，Ｉ／Ｏ記述部２，Ｉ／Ｏデータ部３に分けた構成とし、ソフトウェア結合手段４によって結合させる。

【効果】アプリケーション部１とＩ／Ｏ記述部２を分けて記述するため、記述ソフトウェアが簡潔になる。またＩ／Ｏデータ部３をＩ／Ｏ記述部２とは独立して設けたのでハードウェアの変更時にＩ／Ｏデータ部３の変更だけで済む。さらに、これらを結合して目的プログラム５を生成させるようにしたため、最適化処理がプログラム化できる。

図 1



1

【特許請求の範囲】

【請求項１】マイクロコンピュータを用いて自動車の制御を行う自動車制御装置のソフトウェア構成において、Ｉ／Ｏ（入出力信号）の処理を実行するＩ／Ｏソフトウェア部がタスク管理及び優先順位等の基本ＯＳ（Operating System）プログラムに含まれることを特徴とする自動車用制御装置。

【請求項２】マイクロコンピュータを用いて自動車の制御を行う自動車制御装置において、少なくとも演算処理装置及び記憶手段を有し、上記記憶手段にＩ／Ｏソフトウェア部，タスク管理及び優先順位を有する自動車用ＯＳ（Operating System）プログラムが記憶されていることを特徴とする自動車用制御装置。

【請求項３】請求項２の自動車用ＯＳはＩ／Ｏソフトウェア部とタスク管理，優先順位等から成る基本ＯＳ（Operating System）部を有し、それぞれ異なる記憶手段のアドレスに記憶されていることを特徴とする自動車用制御装置。

【請求項４】マイクロコンピュータを用いて自動車の制御を行う自動車制御装置において、少なくとも演算処理装置及び記憶手段を有し、上記記憶手段に自動車制御用ソフトウェアプログラムが記憶されていることを特徴とする自動車用制御装置。

【請求項５】マイクロコンピュータを用いて自動車の制御を行う自動車制御装置において、Ｉ／Ｏソフトウェア部とタスク管理，優先順位等を含む自動車用ソフトウェアがカタログあるいは仕様書等に記載されていることを特徴とする自動車用制御装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、自動車用制御装置に係り、特に複雑なソフトウェアの開発時や、ソフトウェアの改編時，あるいは使用ハードウェアの変更時に，信頼性高く、かつ少ない作業工数で目的のソフトウェアを得ることができるようなソフトウェア構成を有する自動車用制御装置及び，その結合方法に関する。

【０００２】

【従来の技術】従来の自動車制御ソフトウェアは，目的とする制御動作が行えるように，制御アプリケーションからハードウェアへの入出力までを，使用するハードウェアのアーキテクチャに合わせて一連のプログラムとして記述を行っていた。

【０００３】従って、プログラム作成の際には，例えばハードウェアについての説明書（ハードウェアマニュアル）を参照しながら，目的の制御動作に適したハードウェアの使用法を検討し、決定された使用形態においてハードウェアが最適な動作を行えるよう，ソフトウェア構成を考え、各タスクへの割付設計を行い、各ジョブの作成を行うといった手順によって，プログラム全体の動きを考慮しながら一連のプログラムを作成していた。また、ハードウェアへの入出力動作記述についても、これら一連のプログラム作成手順の中で同時に検討され、リアルタイム制御がうまく実現されるようプログラム中の好適な位置に、アプリケーションと混在する形で記述され、その記述内容についても、プログラム作成者が，使用するハードウェアについてのハードウェアマニュアルを理解し、使用するハードウェアの流儀に合わせて動作手順，内容を記述していた。

2

【０００４】【発明が解決しようとする課題】しかしながら、自動車制御用のソフトウェアを開発するにあたり，このようなプログラムの作成法によれば、プログラム作成時において、プログラム作成者は本来記述したい制御アプリケーションの内容と共に、使用するハードウェアに合わせた形で、ハードウェアに対する信号の入出力処理についても記述しなければならない。したがって、プログラム作成者には、自動車制御の知識の他に、例えばマイコンのアーキテクチャや、操作法など、ハードウェアに関する多くの知識を必要とした。

【０００５】また、自動車の制御に好適なリアルタイム制御を実現するためには、演算，入出力等の各処理が最適なタイミング、および手順で行われる必要があり、プログラムの作成にあたっては、これらが全体にわたってうまく動作するように良く考えなくてはならない。したがって、このようなプログラムの作成法によれば、プログラム作成者はプログラム全体の流れを常に把握していなければならず、複雑なプログラムの作成時や、アプリケーションの変更時には多大な労力を必要とした。特に、複数のプログラム作成者が共同で作業を行う場合にはこの傾向は顕著にあらわれ、作成者全員がプログラム全体の流れを理解しなければならないことになり、きわめて非効率的であると同時に、注意微漫によるミスを誘発するものとなっていた。

【０００６】さらに、自動車制御装置のハードウェア条件（コスト，ＲＯＭ，ＲＡＭ容量）は厳しく、その上リアルタイム制御であることからその応答速度に対する要求も厳しく、これらすべての条件を満たすようなプログラムを作るためにはプログラムの最適化の作業が欠かせないが、このようなプログラムの作成法ではプログラムの変更のたびにこのような最適化を行う必要があり、作業効率上からも好ましくないし、プログラム作成者が交代するたびに、あらためてノウハウを会得していかなければならない点もプログラムの生産効率を下げる要因となっていた。

【０００７】本発明の目的は、プログラム作成時の前記問題点を解決し、効率の良い分業を可能にし、アプリケーション部分の作成に十分な検討が可能で、かつ全体としては少ない作成工数で、信頼性の高いプログラムの作成が可能となるとともに、制御アプリケーションの変更やハードウェアの変更などに対しても汎用性のある自動

車制御用ソフトウェア構成、およびそれらの結合処理方法を提供するものである。

【0008】

【課題を解決するための手段】前記目的を達成するために、本発明では、制御用のソフトウェアに、制御内容を記述したアプリケーション部と、ハードウェアに対する入出力処理について記述したＩ／Ｏ記述部２とを設け、これらと、Ｉ／Ｏデータ部３に示されたハードウェアに関する情報とを、結合手段を用いて結合し、目的のプログラムを得るようにした。

【0009】

【作用】このように構成され、処理される本発明によれば、複雑な自動車制御ソフトウェアの作成時にも、アプリケーション部とＩ／Ｏ記述部２を分けたために、ハードウェアに対する入出力動作に関するソフトウェア設計と、制御のための演算処理動作に関する設計とを別々に行うことができる。さらに、それぞれが別々に設計されていることから、アプリケーションや入出力処理が変更された場合にも記述の一部を修正し、結合することによって、容易に目的のプログラムを得ることができる。

【0010】また、ハードウェアに対する入出力処理内容をＩ／Ｏ記述部２に標準化された表現を用いて記述し、ハードウェア自身に関するデータをＩ／Ｏデータ部３にそれぞれ分離して記述するようにしたことから、ハードウェアの変更時でも、ハードウェアに付随してＩ／Ｏデータ部３を差し替えればその他の部分はそのまま使用する事ができる。

【0011】さらに、別々に作った各部を、結合手段によって結合して目的のプログラムを生成させる方式としたことから、結合手段にプログラムの最適化処理のノウハウを記憶させることができる。

【0012】

【実施例】以下、本発明の実施例を図面に基づき詳細に説明する。

【0013】図１は本発明の第１実施例である。図において、ソフトウェア開発時に各プログラム作成者、またはソフトウェア開発者によって記述または改編される記述ソフトウェア６には、アプリケーション部１，Ｉ／Ｏ記述部２，Ｉ／Ｏデータ部３が設けられている。このアプリケーション部１には、制御において、制御対象の動作内容、およびそのための各値を決定するための演算や論理処理などの、制御の本質的な内容が記述される。また、Ｉ／Ｏ記述部２には前述のアプリケーション部１で必要とされるハードウェアの各値または信号をソフトウェア上に取り込み、あるいはアプリケーション部１において演算，論理処理された結果の制御値や動作内容をハードウェアへ出力する各処理などの、ハードウェアに対する入出力処理、およびその処理タイミングが一般化された表現で記述されている。さらにＩ／Ｏデータ部３には、ハードウェアの構成や機能，その設定方法，アドレスデータなどの一連の実データなど、従来ハードウェアマニュアルに記載されていた事項がアプリケーション部１やＩ／Ｏ記述部２で使用しやすい形にまとめられて記載される他に、制御対象とマイコン端子の接続など、システム全体の構成に関する事項や、目的の制御に特有のハードウェア操作を容易に実現するためのデータ，ハードウェアとソフトウェア間の使用に関する取り決めから自動的に決まるデータなどが記載されており、各制御内容の記述を容易にし、定例的な事項についての記述を省略可能とする事で、プログラム記述を容易にしている。なお、これらのデータは後述するソフトウェア結合手段５において読み出され、利用できるように整理された形で記載されている。

【0014】これらの記述ソフトウェアの各部分は、記述後、ソフトウェア結合手段４によって結合され、ハードウェア上で実際に制御動作を行う目的プログラム５が生成される。このとき、ソフトウェア結合手段４は、そのソフトウェアを搭載するハードウェアに対応した各データＩ／Ｏデータ部３に記載されたデータを参照しながらＩ／Ｏ記述部２を展開しつつ、アプリケーション部１とのリンクを行うことによって目的のプログラムを生成させるとともに、プログラムの最適化を行い、無駄処理を取り除き、効率の良い目的プログラム５を作り出す。

【0015】ここで、このソフトウェア結合手段４は、前述のアプリケーション部１，あるいはＩ／Ｏ記述部２，あるいはＩ／Ｏデータ部３に含まれても良いし、あるいはこれら３つとは独立して設けられていても構わない。

【0016】前述のように、従来ハードウェアマニュアルの形で提供されていた、マイコン等のハードウェアに関する情報を、ユーザーの使用形態に合わせた形に整理し、データベース化した形のユーザーインターフェース型ハードウェアマニュアルとして提供することで、ユーザーは独自の表現、および方法でソフトウェアの開発をする事が可能となり、同時に、ハードウェアの変更を行った場合でも、付属のユーザーインターフェース型ハードウェアマニュアルを用いることで、きわめて少ない作業工数でソフトウェアをそのまま利用することが可能となる。

【0017】図２は本発明の第２実施例である。図において、記述ソフトウェア２５には制御の本質的内容が記述されたアプリケーション部２１，ハードウェアに対する入出力動作のほかにハードウェアの構成，機能，使用法など、入出力処理およびハードウェア関連データの記載されたＩ／Ｏソフトウェア部２２が設けられている。

【0018】これらは、ソフトウェア結合手段２４によって結合され、最適化されてハードウェア上で実際に制御動作を行う目的プログラムが生成される。ただし、このソフトウェア結合手段は、前述のアプリケーション部２１、あるいはソフトウェア記述部２２に含まれても良

いし、あるいはこれら２つとは独立して記述されていても構わない。

【0019】Ｉ／Ｏソフトウェア部２２には、ユーザーの希望するＩ／Ｏ動作仕様またはアプリケーション部に記述された入出力処理の命令に応じて、使用ハードウェアに適合した入出力動作を行うようなソフトウェアが記述されるため、ハードウェアの変更に際しては、変更後のハードウェアに対応したＩ／Ｏソフトウェア部２２に差し替えることでアプリケーション部２１はほとんど変更せずに使用できるため、ソフトウェアの汎用性が高くなる。

【0020】図３は本発明の第３実施例である。図において、プログラム部３１には制御の本質的内容を記述したアプリケーション、および、ハードウェアに対する入出力処理動作を一般化した表現で記述したＩ／Ｏ処理内容が記述される。また、Ｉ／Ｏデータ部３２には、ハードウェア構成，機能，使用法など、ハードウェアおよびその利用法に関するデータが後述のソフトウェア結合手段において利用できるような形に整理されて記載されている。

【0021】ソフトウェア結合手段３３は前記Ｉ／Ｏデータ部３２に記載されたハードウェアに関するデータや利用法，プログラム記述に関する申し送り事項などを参照しながらプログラム部を展開し、最適化処理を行って、ハードウェア上で実際に制御動作を行う目的プログラム３４を生成する。

【0022】ここで、このソフトウェア結合手段は、前述のプログラム部３１，あるいはＩ／Ｏデータ部３２に含まれても良いし、あるいはこれら２つとは独立して設けられていても構わない。

【0023】図４は前記図１で示した第１実施例について、ソフトウェアの結合方法の一実施例を示したものである。図にしめされるたように、各ソフトウェア作成者によってそれぞれ記述されたアプリケーション部４１，Ｉ／Ｏ記述部４２，Ｉ／Ｏデータ部４３は、ソフトウェア結合手段１４４およびソフトウェア結合手段２４６によって結合されるが、その順序は以下に示すとおりである。

【0024】まず第一に、アプリケーション部４１とＩ／Ｏ記述部４２がソフトウェア結合手段１４４によって結合され、プログラム部４５が生成される。このとき、結合手段１４４のもつソフトウェア構成のノウハウを利用して、演算などのアプリケーション処理動作とハードウェアに対する入出力処理動作のそれぞれについて動作タイミング、および処理順序の検討が行われ、目的の制御動作に適したプログラム構成となるよう最適化が行われる。次に、ソフトウェア結合手段２４６は、Ｉ／Ｏデータ部４３を参照しながら、先に生成されたプログラム部４５のＩ／Ｏ処理動作内容の記述を使用ハードウェアの仕様に合わせた書き方に変換し、目的プログラム４７を得る。

【0025】図６は前記図１で示した第１実施例について、ソフトウェアの結合方法、およびハードウェアへの搭載方法についての一実施例を示したものである。図にしめされるように、記述ソフトウェア６９にはアプリケーション部６１、およびＩ／Ｏ記述部６２が設けられ、これらは結合手段６４によって結合され、生成プログラム６５が生成される。また、Ｉ／Ｏデータ６３はＲＯＭ６７上などに記録された形で提供され、制御装置上６６の前記生成プログラム６５が必要に応じて呼び出して使用する。

【0026】ただし、生成プログラム６５の持つＩ／Ｏデータ６３利用機能については、アプリケーション部６１、またはＩ／Ｏ記述部６２にあらかじめ含まれていても良いし、あるいはソフトウェア結合手段６５によって結合時に付加されても良い。前記図１から図６において示してきたソフトウェア構成および結合方法を用いることによって、ソフトウェアの記述が煩雑になることに加えて、各制御アプリケーションおよびＩ／Ｏ記述をそれぞれ別々に設計し、結合することが可能となるので、多数の制御項目をもつ複雑な制御アルゴリズムの記述が容易となり、また、Ｉ／Ｏデータを活用することによって、共通のハードウェアをさまざまな用途に、かつ使いやすい形態で提供することが可能となる。

【0027】例えばエンジン制御，ＡＴ制御のような複雑な制御ソフトウェアも、いくつかの小さなブロックに分けて記述することにより、極めて簡潔に制御内容を記述できる。その一例として以下に簡単なエンジンの点火制御モデルの記述を示す。

【0028】図７は説明するエンジン制御モデルのハードウェアシステム構成を示したものである。

【0029】エンジンの点火は図に示されたプラグ１～６　７４のギャップ間放電によって行うが、そのために、コイル７３に一定時間電流を流した後、電流を切断する制御が行われ、電流が切断された瞬間、コイルに蓄えられた誘導電流のエネルギーがギャップ間の放電の形で放出される。したがって、コイル７３の電流がonしてからoff するまでの時間（以下通流時間）の制御は、点火タイミング、すなわち電流がoff するタイミングの制御と共に極めて重要なものである。

【0030】なお、本モデルにおいて、制御用のマイコン７１にはエンジンのクランク角の基準位置を知らせるためのＲｅｆ信号７６，クランク角の回転を知らせるＰＯＳ信号７７，および吸入空気温度Ｔｗ７８，冷却水温度Ｔｗ７９などが入力されており、Ｒｅｆ信号７６は１２０度おきに、ＰＯＳ信号７７は１度おきに入力される。また、ＩＧＮ信号出力端子７５からはプラグ７４への制御信号が出力されるが、この出力においては、約１２０度おき点火制御が行われる。実際にはエンジンは２回転に１回だけ点火すればよいので、この出力信号は分

配器７２によって６つに分配され、６気筒エンジンでは、各気筒がちょうど７２０度に１回点火するようになっている。

【００３１】図８はマイコン８６内のハードウェア構成を示したもので、本モデルで使用するタイマー８９とＣＰＵ８５が示されている。タイマー８９にはタイマーカウンタ８１や比較器８２，コンペアレジスタ８３，タイマーコントロールレジスタ８４があり、タイマーカウンタ８１には前述のＰＯＳ信号が入力され、クランク角が１度回転するたびにカウンタの値も１ずつ増えるようになっている。コンペアレジスタ８３の値とタイマーカウンタ８１の値は常に比較器８２によって比較され、２つが一致した時にはタイマー出力８７を変化させたり、ＣＰＵ８５に割り込み出力信号８８を発生する事ができる。したがってタイマーを使って点火制御を行うには、タイマーの機能を選択するためのタイマーコントロールレジスタ８４に必要な設定となるように値を書き込むとともに、動作を起こしたい時点のタイマーカウンタの値を演算し、ＣＰＵからコンペアレジスタ８３に書き込めばよい。

【００３２】なお、Ｒｅｆ信号はＣＰＵ８５に割り込みの形で入力される。

【００３３】図９は本エンジン点火制御のタイムチャートを示したものである。制御したい内容は、「Ｒｅｆ信号の立ち上がり９７からａだけ過ぎた時点Ｙ９６でＩＧＮ信号をoff(点火)する」事と、「コイルの通流時間をｂにする」事であるが、後者は記述しにくいので「点火時点Ｙ９６からｂだけ手前の時点Ｘ９５で出力をonにする」と記述することにする。

【００３４】ここで、前述のコンペアレジスタには動作を起こしたい時点のカウンタ値を代入しておくことになっているので、たとえば「Ｒｅｆ信号の立ち上がり９７からａだけ過ぎた時点Ｙ９６」をセットするには、まずＲｅｆ立ち上がり９６時のカウンタ値Ｓ９４を読み込まなくてはならない。そして、コンペアレジスタに代入する値はｓ＋ａである。

【００３５】図１０に本モデルを前記図１に示したソフトウェア構成により記述した一実施例を示す。図において、結合前のソフトウェアはアプリケーション部１０１，Ｉ／Ｏ記述部１０２，Ｉ／Ｏデータ部１０３の３部から構成され、これらは前記図４に示したように、まずアプリケーション部１０１とＩ／Ｏ記述部１０２が結合され、ついで、得られたプログラムをＩ／Ｏデータ部１０３を参照しながら展開するという手順で目的のプログラムを得る。

【００３６】ここで、各部の内容について説明する。アプリケーション部において@に続く一節は、その起動タイミングにおいて、処理を行うことを示す。したがってアプリケーション部１０１の内容は、まず、Ｒｅｆ信号が立ち上がった時点（Ｒｅｆ割り込み時）に、演算Ｘ＝Ｓ＋ａ－ｂと演算Ｙ＝Ｓ＋ａを行うことを意味している。ついで@ＢＧＪの起動タイミングでＸ，Ｙを求める演算を行っているが、@ＢＧＪとはバックグラウンド、すなわち「他の処理をしていないとき」、すなわち空き時間にこの処理を繰り返し行うことを意味している。

【００３７】Ｉ／Ｏ記述部１０２に記述される処理は、上から順に、「Ｒｅｆ割り込み時にＩＧＮ（点火用の意味）というグループ名のタイマーカウンタ値を読み込み、変数Ｓに代入する」、「Ｒｅｆ割り込み時に、ＩＧＮというグループ名のタイマーの値がＸのときon信号を出力するようにセット処理する」、「ＩＧＮ出力がonに変わった時に、ＩＧＮというグループ名のタイマーの値がＹのときoff 信号を出力するようにセット処理する」ということを意味している。

【００３８】これらを連続的に行えば、前記図９で示したような目的の制御が実現できる。アプリケーション部１０１とＩ／Ｏ記述部１０２の結合においては、基本的には以下の順序にしたがって組み替えを行う。すなわち、同じ起動タイミングで定義された処理については、Ｉ／Ｏ処理のデータ読み込み，アプリケーションの処理，Ｉ／Ｏ処理の各種書き込みの順に組み替える。

【００３９】このような操作が前記図４で示した結合手段１によって行われるのに次いで、Ｉ／Ｏデータ部３を参照しながら、図４における結合手段２による処理が行われる。

【００４０】ここで、図１２に示したＩ／Ｏデータについて説明する。データはテーブル状に表現されており、記載情報の種類１２１，データ１２２，プログラム展開時の使い方１２３についての情報が記載されている。このプログラム展開時の使い方１２３はソフトウェア結合手段が様々な形で利用できるように対応した形で記載されており、その内容は、ハードウェアのデータ１２４，構成システムのデータ１２５，ユーザーの使用形態に合わせて記載されるソフトウェアへの引き渡しデータ１２６，通例的に使用することの多い制御上のデータ１２７などで、ハードウェアへの対応の他、ソフトウェア開発者の利便性の向上やプログラムの最適化に必要なデータがデータベース化されている。

【００４１】このようなＩ／Ｏデータを参照しながら本点火制御プログラムをＣ言語に展開した目的プログラムを図１３に示した。

【００４２】図１０において@Ｒｅｆのついていた処理内容は図１３（－ｃ）のRefJobに登録されており、別々に記述された処理が、結合され、１つのプログラムを形成している。

【００４３】ここで、図１３（－ａ）のinitiallizationに登録された処理内容等は前述の図１０における表記には直接的にはみられないもので、Ｉ／Ｏデータおよび結合手段から生成したものである。

【００４４】図１１は前記図１０と同様の制御内容を、

前記図３に示したソフトウェア構成によって記述したもので、同様に図１２のＩ／Ｏデータを用い、展開後には前記図１３に示した目的プログラムを得るが、前記図１０の場合と比べて、ソフトウェア結合の為の処理が１段階減っているのが特徴である。

【００４５】図１４は本発明のソフトウェア構成を自動車用シングルチップマイコンに展開した場合の一実施例である。ソフトウェア構成としては、アプリケーション部１，Ｉ／Ｏソフトウェア部２２及び自動車用ＯＳ（Operating System）部１０１の３つに分けられる。Ｉ／Ｏソフトウェア部２２は前述のようにＩ／Ｏ記述部２及びCPU104（中央演算処理装置）毎に異なるＩ／Ｏデータ部３をソフトウェア結合手段１００に入力し作成する。そして、上記Ｉ／Ｏソフトウェア部２２及び自動車用ＯＳ部１０１をシングルチップマイコン１０２に記憶手段１０３（ROM：読み出し専用メモリ、E²PROM：電気的に書換え可能なメモリ等）に記憶させユーザーに提供する。ユーザーはアプリケーション部１を作成し、上記記憶手段１０３に書き込み制御を実行する。これにより、例えば、シングルチップマイコン１０２の性能及びハード構成が変更になった場合、更には、Ｉ／Ｏ構成が変更になった場合でも、ユーザーはアプリケーション部１を移植することができ、アプリケーションソフトをリピータビリティが向上する。この時、マイコンメーカはＩ／Ｏデータ部３を変更して自動車用ソフトウェア付きシングルチップマイコン１０２を提供すれば良い。また、プログラムはマイコンのハードを意識せずアプリケーションソフトを作成することができる。

【００４６】図１５はソフトウェア構成を自動車用ＯＳプログラムとした場合の一実施例である。ここでは、上記Ｉ／Ｏソフトウェア部２２とタスク管理及び優先順位等から成るタスクディスパッチャの基本ＯＳ１０５を用いて自動車用ＯＳ１０６を作成する。このＯＳは自動車専用であり、自動車制御用として汎用性もある。また、図１４と同様にシングルチップマイコン１０２の記憶手段１０３に書き込んで提供することも可能である。

【００４７】

【発明の効果】本発明によれば、アプリケーション部とＩ／Ｏ記述部２を分けて記述するようにしたために、ハードウェアに対する入出力動作に関するソフトウェア設計と、制御のための演算処理動作に関する設計とを別々に行うことができるので、複雑なソフトウェアの設計が簡潔になり、プログラム作成工数が低減する共に、プログラムの信頼性を上げることができる。

【００４８】また、目的のプログラムを得るのに、アプリケーション部とＩ／Ｏ記述部２を分けて記述し、それらを結合する方法を用いたために、アプリケーションや入出力処理が変更された場合にも記述の一部を修正し、結合することによって、容易に目的のプログラムを得ることができるので、ソフトウェア汎用性が高くなる。さらに、ハードウェアに対する入出力処理内容をＩ／Ｏ記述部２に標準化した表現を用いて記述し、ハードウェアに関するデータをＩ／Ｏデータ部３にそれぞれ分離して記述するようにしたことから、ハードウェアの変更時でも、ハードウェアに付随してＩ／Ｏデータ部３を差し替えればその他の部分はそのまま使用する事ができるので、ハードウェアソフトウェアの汎用性が高くなる。

【００４９】さらに、別々に作った各部を、結合手段によって結合して目的のプログラムを生成させる方式としたことから、結合手段にプログラムの最適化処理のノウハウを記憶させることができるので、プログラム作成者は特に意識することなしに、プログラム作成技術の継承ができる。

【図面の簡単な説明】

【図１】本発明による、アプリケーション部，Ｉ／Ｏ記述部２，Ｉ／Ｏデータ部３に分けて記述されたソフトウェアを結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

【図２】本発明による、アプリケーション部，Ｉ／Ｏソフトウェア部に分けて記述されたソフトウェアを結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

【図３】本発明による、プログラム部，Ｉ／Ｏデータ部３に分けて記述されたソフトウェアを結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

【図４】本発明による、別々に記述されたアプリケーション部，Ｉ／Ｏ記述部２を結合し、さらにＩ／Ｏデータ部３を結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

【図５】本発明による、別々に記述されたＩ／Ｏ記述部２，Ｉ／Ｏデータ部３を結合し、さらにアプリケーション部と結合することによって目的のプログラムを得ることを示したソフトウェア作成方法を表す概念図。

【図６】本発明による、アプリケーション部，Ｉ／Ｏ記述部２を結合した生成プログラムを制御装置上に、またＩ／Ｏデータ部３をデータ用ＲＯＭに載せ、データを参照しながらプログラムを動作させることを示した構成図。

【図７】エンジンの点火制御を行うに必要なハードウェア構成の説明図。

【図８】タイマーを含むマイコンの動作を説明するための構成図。

【図９】点火制御モデルの動作を説明するためのタイムチャート。

【図１０】点火制御をアプリケーション部，Ｉ／Ｏ記述部２，Ｉ／Ｏデータ部３に分けて記述した例を示すプログラム図。

【図１１】点火制御をプログラム部，Ｉ／Ｏデータ部３に分けて記述した例を示すプログラム図。

11

【図１２】ハードウェアデータを結合手段が利用できる形に整理して示したＩ／Ｏデータテーブル図。

【図１３】本発明による分離記述を行ったソフトウェアが、結合処理によって一連のソースプログラムとなることを示したプログラム図。

【図１４】本発明のソフトウェア構成を自動車用シングルチップマイコンに展開した場合の一実施例を示す図。

12

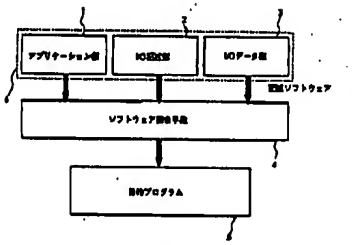
【図１５】本発明のソフトウェア構成を自動車用ＯＳプログラムとした場合の一実施例を示す図。

【符号の説明】

１…アプリケーション部、２…Ｉ／Ｏ記述部、３…Ｉ／Ｏデータ部、４…ソフトウェア結合手段、５…目的プログラム。

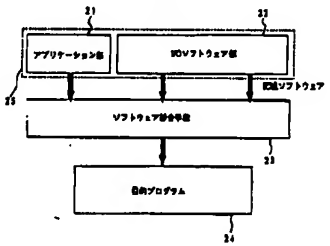
【図１】

図 1



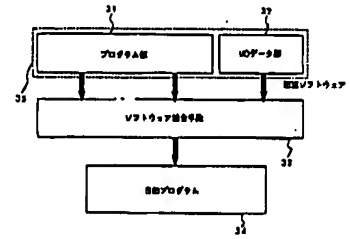
【図２】

図 2



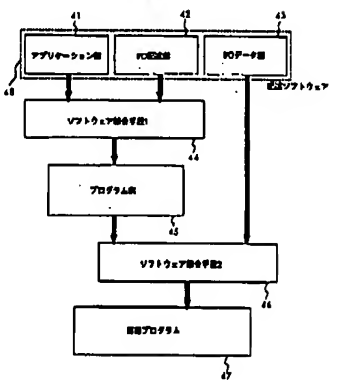
【図３】

図 3



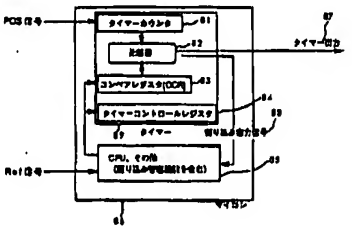
【図４】

図 4



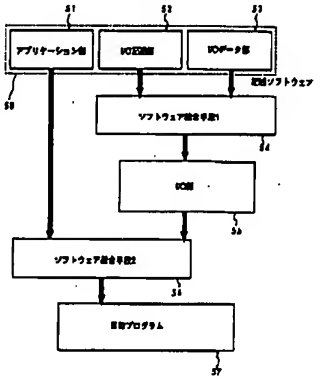
【図８】

図 8



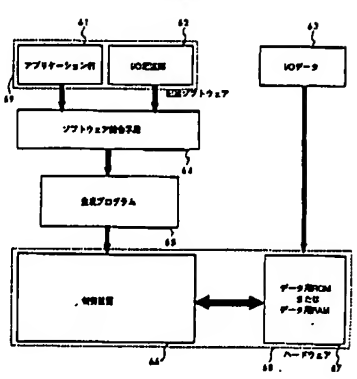
【図５】

図 5



【図６】

図 6



【図７】

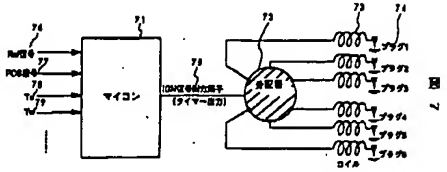


図 7

【図９】

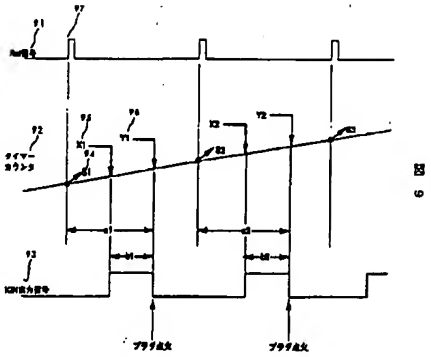
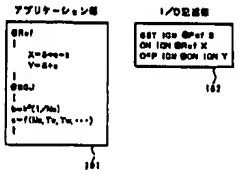


図 9

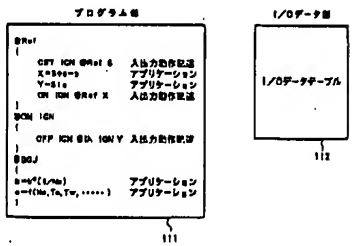
【図１０】

図 10



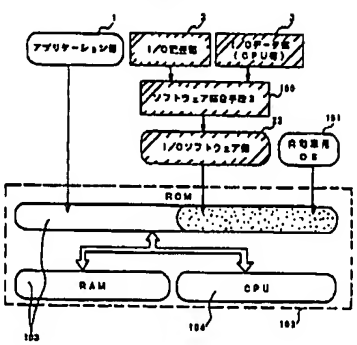
【図１１】

図 11



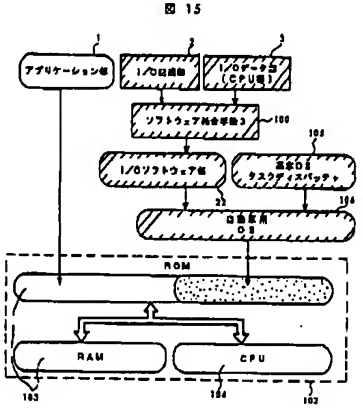
【図１４】

図 14



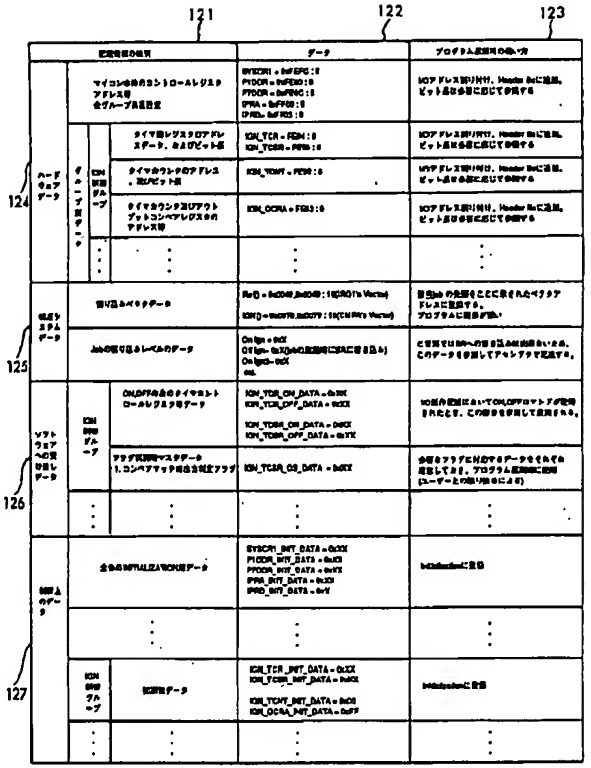
【図１５】

図 15



【図１２】

図 12



【図13】

図 13

(a) Initializationへ登録

```
char DUMMY;
Init_Ign()
{
   IGN_TCR = IGN_TCR_INIT_DATA;
   IGN_TCSR = IGN_TCSR_INIT_DATA;
   SYSCR1 = SYSCR1_INIT_DATA;
   SYSCR1 = SYSCR1_INIT_DATA;
   P1DDR = P1DDR_INIT_DATA;
   P7DDR = P7DDR_INIT_DATA;
   IPRA = IPRA_INIT_DATA;
   IPRD = IPRD_INIT_DATA;
}
```

(b) Background Jobへ登録

```
On_Ign()
{
   b = k * (1/Ne);
   a=f(Ne,Tw,Te,...)
}
```

(c) Ref Jobへ登録

```
Ref_On_Ign()
{
   register char S = IGN_CNT;
   X=S-a-b;
   Y=S+a;
   #pragma asm
   STC.W SR,@-SP
   LDC.W #SEI6,SR
   #pragma endasm
   if(X>IGN_CNT)
   {
      IGN_OCRA = X;
      IGN_TCR = IGN_TCR_ON_DATA;
      IGN_TCSR = IGN_TCSR_ON_DATA;
   }
   #pragma asm
   LDC.W @SP+,SR
   #pragma endasm
```

フロントページの続き

(72)発明者 石井 潤市
茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(72)発明者 森永 茂樹
茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内